**УО «Белорусский государственный университет информатики и радиоэлектроники»**

**Кафедра ПОИТ**

**Отчёт по лабораторной работе №2.2**

**По предмету**

**Основы алгоритмизации и программирования**

**Вариант 6**

**Выполнил:**

**Гладкий М.Г.**

**Проверила:**

**Данилова Г.В.**

**Группа 851001**

**Минск 2018**

**Задание:**

Если последнюю цифру некоторого натурального числа n перенести и поставить перед первой цифрой этого числа, то получится число, в два раза больше n. Найти самое маленькое из таких чисел.

**Код Delphi 10:**

**program** Project6;

{$APPTYPE CONSOLE}

**type**

TArr = **array** [2..8] **of** Int64;

**function** FindNumber(i: Byte): Int64;

**var**

Number, Prev, Step: Int64;

Digit: Byte;

**begin**

Prev := 0;

Digit := i;

Number := i;

Step := 1;

**repeat**

Step := Step \* 10;

Number := Number + Step \* (Digit \* 2 **mod** 10 + Prev);

Prev := Digit \* 2 **div** 10;

Digit := Number **div** Step;

**until** (Digit = i) **and** (Prev = 0);

Number := Number - Digit \* Step;

FindNumber := Number;

**end**;

**function** GetArray(): TArr;

**var**

i: Byte;

MyArr: TArr;

**begin**

Writeln('This program searches for the smallest number, which, when you

rearrange the last digit to the first one, is twice as large as

the original');

Writeln('This is the list of required numbers :');

**for** i := 2 **to** 8 **do**

**begin**

MyArr[i] := FindNumber(i);

Writeln(i,' : ',MyArr[i]);

GetArray[i] := MyArr[i];

**end**;

**end**;

**procedure** ShowAnswer(BuferArray: TArr);

**var**

Min: Int64;

i: Byte;

**begin**

Min := BuferArray[8];

**for** i := 2 **to** 8 **do**

**if** BuferArray[i] < Min **then**

Min := BuferArray[i];

Writeln('Minimum required number is : ', Min);

Readln;

**end**;

**procedure** Main();

**var**

NumArr: TArr;

**begin**

NumArr := GetArray();

ShowAnswer(NumArr);

**end**;

**begin**

Main;

**end**.

**Код С:**

#include <stdio.h>

struct TArr

{

\_\_int64 MyArray[7];

};

\_\_int64 FindNumber(int i)

{

\_\_int64 Number = i, Prev = 0, Step = 1, Digit = i;

do

{

Step \*= 10;

Number += Step \* ((Digit \* 2) % 10 + Prev);

Prev = (Digit \* 2) / 10;

Digit = Number / Step;

} while ((Digit != i) || (Prev != 0));

Number -= Digit \* Step;

return Number;

}

struct TArr GetArray()

{

\_\_int64 MyArr[7];

printf\_s("This program searches for the smallest number, which, when you rearrange

the last digit to the first one, is twice as large as the original\n");

printf\_s("This is the list of required numbers :\n");

for (int i = 0; i < 7; i++)

{

MyArr[i] = FindNumber(i + 2);

printf\_s("%d : %I64d\n", i + 2, MyArr[i]);

}

struct TArr MyArray = { MyArr[0], MyArr[1], MyArr[2], MyArr[3], MyArr[4],

MyArr[5], MyArr[6] };

return MyArray;

}

void ShowAnswer(TArr MyArray)

{

\_\_int64 Min = MyArray.MyArray[6];

for (int i = 0; i < 7; i++)

if (MyArray.MyArray[i] < Min)

Min = MyArray.MyArray[i];

printf\_s("Minimum required number is : %I64d", Min);

}

int main()

{

TArr NumArr = GetArray();

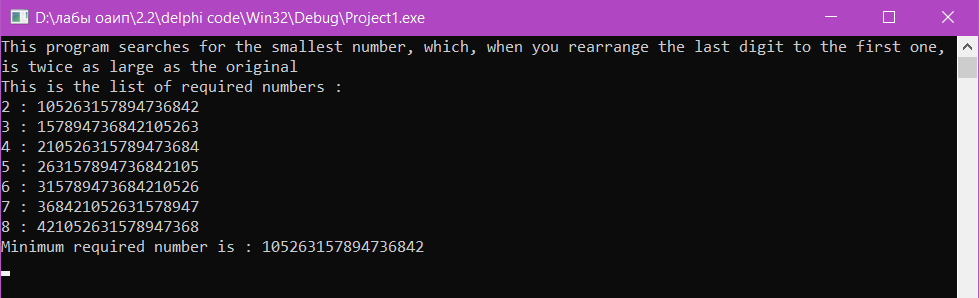
ShowAnswer(NumArr);

return 0;

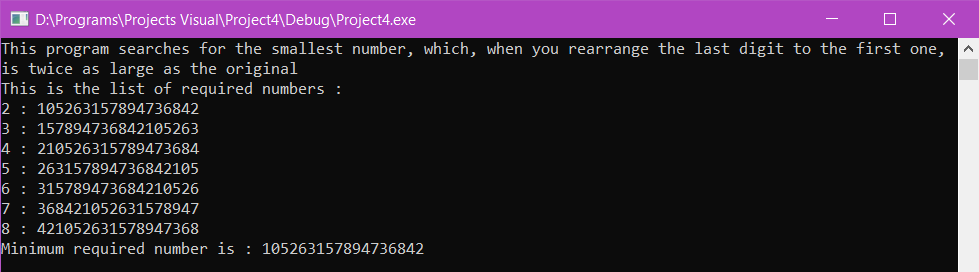
}

**Скриншоты:**

**Delphi 10:**



**C:**



**Блок-схема:**

